

基于音高旋律元的柔和乐曲的自动作曲算法

曹西征¹ 毛文涛¹ 乔锬¹ 程小丽¹ 蔡会娟¹ 温辉明²

摘要 提出了一种以柔和为导向、以音高旋律元 (Pitch melody unit, PMU) 为基本操作单位的自动作曲算法。首先根据柔和乐曲的旋律特征构建了算法的规则模型。然后定义了一阶音高旋律元和二阶音高旋律元, 构建了音高旋律元转换表 (Pitch melody unit transition tables, PMUTTs) 自动生成的模型, 以表达相邻音高旋律元的链接概率, 为旋律计算提供基本数据。分别定义了用来自动生成新的音高旋律元、乐节、乐句、乐曲的 4 个旋律计算的算子, 以自动生成乐曲。实验结果表明, 与以音符为基本单位的马尔可夫作曲方法相比, 以音高旋律元为基本单位进行作曲可以提高旋律的逻辑严密性和美感度。

关键词 自动作曲, 音高旋律元, 柔和乐曲, 规则模型, 旋律计算

引用格式 曹西征, 毛文涛, 乔锬, 程小丽, 蔡会娟, 温辉明. 基于音高旋律元的柔和乐曲的自动作曲算法. 自动化学报, 2012, 38(10): 1627–1638

DOI 10.3724/SP.J.1004.2012.01627

Automated Composition Algorithm for Gentle Music Based on Pitch Melody Unit

CAO Xi-Zheng¹ MAO Wen-Tao¹ QIAO Kun¹ CHENG Xiao-Li¹ CAI Hui-Juan¹ WEN Hui-Ming²

Abstract In this paper, an automated music composition algorithm for gentle music is proposed based on pitch melody unit (PMU) which is the minimal operation unit. Firstly, several algorithmic rule models are constructed according to the melody characteristic of gentle music. Secondly, the 1st-order and 2nd-order PMUs are defined, and a model is further constructed to automatically generate pitch melody unit transition tables (PMUTTs) which are the expression of link probability of two adjacent PMUs. Then, the basic data for melody computing are obtained. Thirdly, four operators of melody computing are defined to generate a new PMU, phrase, sentence, and music, respectively. Finally, the gentle music can be produced automatically. Experimental results demonstrate that this PMU-based approach can make the obtained melody more logical and aesthetic than the Markov model approach which is based on operating notes.

Key words Automated composition, pitch melody unit (PMU), gentle music, rule model, melody computing

Citation Cao Xi-Zheng, Mao Wen-Tao, Qiao Kun, Cheng Xiao-Li, Cai Hui-Juan, Wen Hui-Ming. Automated composition algorithm for gentle music based on pitch melody unit. *Acta Automatica Sinica*, 2012, 38(10): 1627–1638

随着现代社会人们对精神文化生活需求的不断增长和文化体制改革的不断深入, 计算机自动作曲也日益成为一个越来越重要的课题。自动作曲的研究不仅对自动化技术、计算机应用、人工智能、多媒体技术、音乐治疗^[1-3]、音乐科技、逻辑学、心理学等理论的发展具有重要的推动作用, 而且在音乐创作的高效性、适应性、多样性等方面具有重要的现

实意义。

自动作曲 (算法作曲) 是试图使用某个形式化的过程, 以使人 (或作曲家) 在利用计算机进行音乐创作时介入程度达到最小的研究^[4-5]。目前在自动作曲中公认的有影响的技术主要有马尔可夫链^[6-9]、随机过程、基于规则的知识库系统、音乐文法、人工神经网络^[10-11]、遗传算法^[12-17] 等, 文献 [4] 对 2005 年前的文献中出现的这些技术进行了详细的分析, 说明了这些技术的特点和不足, 并概括总结了自动作曲研究中存在的主要问题: 音乐的知识表达问题、创造性和人机交互性问题、音乐创作风格问题, 以及系统生成作品的质量评估问题。

2005 年以来在自动作曲方面发表的论文中又出现了其他技术, 例如分层结构^[18]、细胞自动机理论^[19-20]、基于经验的作曲^[18]、回答集编程^[21]、可能性构造空间思维模型^[22]、概率逻辑^[23] 等。分层结构又被称为“无限细节结构”, 它是利用自相似和噪声特性作曲, 从较高抽象的层次来理解音乐与听觉的关系。分层结构更多地可以以检验而不是生成算法

收稿日期 2011-12-15 录用日期 2012-04-09
Manuscript received December 15, 2011; accepted April 9, 2012
国家自然科学基金 (61173071), 河南省重点科技攻关项目 (102102210177, 122102210054), 河南省高校青年骨干教师项目 (2011GGJS-061) 资助
Supported by National Natural Science Foundation of China (61173071), the Key Scientific and Technological Project of Henan Province (102102210177, 122102210054), and the Young Core Instructor Project from the Higher Education Institutions of Henan Province (2011GGJS-061)
本文责任编辑 宗成庆
Recommended by Associate Editor ZONG Cheng-Qing
1. 河南师范大学计算机与信息工程学院 新乡 453007 2. 河南师范大学帕瓦罗蒂音乐艺术中心 新乡 453007
1. College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007 2. Pavarotti Music Art Center, Henan Normal University, Xinxiang 453007

的角色加入到作曲系统之中^[18]. 细胞自动机理论^[19-20]实际上是通过进化规则进行音乐创作, 给出一个细胞和它的邻居在时间 t 时的状态, 规则会决定在时间 $t + 1$ 时细胞的状态, 使用简单的映射并通过细胞自动机理论生成所有的音乐观念. 当然这样的结果很难体现已有音乐作品的风格^[18]. 基于经验的作曲是从作曲者自身的创作经验出发, 建立灵感模型, 根据灵感的形成过程对其进行形式化, 由灵感产生效能, 通过灵感扩展的方式进行作曲^[18], 还需进一步解决的问题主要有经验模型中其他部分的建模问题以及音符多样性问题. 文献 [21] 提出了回答集编程 (Answer set programming) 法, 即使用回答集来建造一个被称为 Anton 的自动化作曲系统, 可生成旋律、节奏、和声, 在即兴演奏、自动生成背景音乐方面可达到实时程度, 为作曲家提供了一个简明的、多功能作曲工具. 此系统能够有效地生成简短的旋律, 但没有解决全局的乐曲结构问题, 需要进一步解决的问题主要有节奏的编码问题以及旋律的规模问题, 文献 [22] 提出了可能性构造空间思维模型, 将创造性思维模型中的可能性构造空间理论应用于音乐作曲领域进行探索与研究, 为智能作曲的研究提供了一个新途径. 文献 [23] 提出了一种用于复调音乐中产生对位的概率逻辑 (Probabilistic logic) 法, 在 Derive 6 下, 对给定的一段固定旋律利用概率算法生成对位, 其结果具有不确定性, 即同一段固定旋律可产生不同的对位. 该算法不考虑旋律的体裁特性, 重点在于解决和声问题, 尚需解决给定不同调式的旋律下的对位生成问题.

自动作曲的研究现状可概括如下:

1) “模仿型”的自动作曲模式占有主导地位. 即“从作品到作品”的模式, 算法的输入为“已有作品”, 输出为“风格仿真的作品”. 这种作曲模式当然有积极意义, 但也存在很大不足: 一是缺乏创造性, 二是以偏概全性 (因为用这单一作品代表要创作的作品风格是片面的, 而且挑选过程也是一项大量耗时的、复杂的工作).

2) “以音符为基本操作单位”的自动作曲模式占有统治地位. 目前现有的作曲算法基本上都是把音符作为最基本的操作单位, 因为它是一种“底层开发”, 创作灵活, 但“过分灵活可能会丢失组织性和纪律性”, 这就导致了即使花费了很大功夫来设计算法, 但结果往往还是不尽如人意, 要想得到满意的结果, 需要更复杂的音符组织, 其计算复杂度会呈指数增长. 倘若把几个音符“预制成一个构件”, 操作这个“构件”会比操作音符这个“砖块”简单一些.

3) 作曲的面向对象往往不够明确. 目前的自动作曲技术侧重于算法本身, 而对由算法生成的作品是面向谁的问题考虑较少, 这种发展模式在一定程

度上使得算法的先进性得到了发展, 然而这种方向性不明确的发展状态会产生“应用不够迫切”的效应, 在另一种程度上阻碍了该项技术的发展.

因此, 只有“一个科学的输入模式、一个有效的操作单位、一个明确的面向对象”作为支撑, 才会使得“一个优秀的计算方法”的效能得到充分发挥.

输入模式和面向对象具有统一性, 它们面临的实际上是同一个问题, 即用户所需作品的风格和它对应音乐要素的特征问题. 因此本文涉及的便是柔和乐曲对应音乐要素的特征问题.

而对于操作单位的有效性, 可结合马尔可夫模型在自动作曲中存在的问题^[4,7]来分析. 马尔可夫模型在自动作曲中存在两种情形: 一是用 2 维的转换表来依次选择音符, 下一个音符是什么只取决于当前的音符和转移概率, 这样会导致由其生成的旋律线条比较随意, 使得旋律的含义不够明确, 因为在旋律中, 下一个音符不仅跟当前的音符有关, 还可能跟前面 1 个或多个音符有关. 二是用 n 维的转换表来选择音符, 这样下一个音符是什么取决于前面 $n - 1$ 个音符, 这与旋律的基本特征相吻合, 但当维数增多时, 转换表的大小呈指数增长, 这就使得它难以管理. 针对这个问题, 本文在考虑算法的基本操作单位时, 着眼于几个音符的音高排列, 以它们的整体作为基本操作单位 (将其定义为音高旋律元), 在构建音高旋律元时以柔和为导向, 这样既可解决 2 维转换表在生成音符的随意性问题, 又可解决 n 维转换表的难以管理问题, 也就是说可用较简单的算法实现有效的自动作曲.

为此, 本文以柔和为目标导向 (因为其无论在现实生活还是在音乐治疗中都是一类备受欢迎的音乐), 以音高旋律元为基本操作单位, 研究柔和乐曲的自动作曲算法. 具体步骤为: 1) 构建算法的规则模型, 对柔和乐曲对应的音乐要素特征进行分析, 然后定义音高旋律元, 构建自动生成旋律元转换表的模型; 2) 以步骤 1) 的结果作为素材, 构建操作这些素材的算法, 定义了 4 个算子; 3) 通过实验对算法进行对比分析.

1 算法的设计规则描述与建模

为了构建算法的规则模型, 设定约束规则, 需要分析柔和乐曲对应的音乐要素特征, 即音乐要素应具备怎样的特征就可以表现“柔和”. “柔和”有“变化缓慢”和“和谐”的意思, 可用“时间变化率小”与“统一性高”来描述. “柔和”是一个触觉和视觉概念, 而音乐是一个听觉概念, 要让一个听觉的东西来表现一个触觉的东西, 需要“联觉”理论的支持.

中央音乐学院周海宏教授在文献 [24] 中以联觉关系为突破口, 用心理学的实证方法证明了与音乐

听觉相关的六种联觉对应关系规律, 以揭示音乐音响与其表现对象之间的对应关系. 本文将在该规律的基础上, 结合作曲理论, 以设计算法为目标来分析“柔和”对应的音乐要素的特征, 构建相应的规则模型.

1.1 音高线

音高线即音的高低随时间律动的线条. 对于柔和乐曲来说, 音高随时间的变化不宜太快, 即音高线斜率的绝对值较小, 即“音高线以级进或小跳为主, 以大跳为辅”. 为了指导算法设计, 构建一个音高律动规则模型 (记为 R_p), 如式 (1)~(3) 所示.

$$\Delta p = |p(i) - p(i-1)| \in S_I \quad (1)$$

$$S_I = \{I | I \geq 0, I \leq I_{\max}, I \in \mathbf{Z}\} \quad (2)$$

$$I_{\max} = \begin{cases} 5, & \text{若 } p(i) \in S_p, p(i-1) \in S_p \\ 12, & \text{若 } p(i) \in S_p, p(i-1) \notin S_p \end{cases} \quad (3)$$

式 (1) 中, $p(i)$ 为第 i 个音符的音高, S_I 为相邻音符的音程的样本集合, 各样本值代表两个音相距的半音个数 (本文中涉及的音程均用半音的个数来表示), 式 (3) 中, S_p 为在知觉统一体内的音高样本集合. 知觉统一体即在人的感觉中属于同一结构之内的东西, 两个音如果不在一个知觉统一体内, 即便相邻, 也不会发生统一感的判断, 或统一感的体验较弱^[24]. R_p 表达的音乐信息为“在知觉统一体之内只使用级进或小跳, 在知觉统一体之间可以使用大跳”.

1.2 速度和节奏

由于柔板与快板相对, 因此柔和乐曲的速度和节奏的变化都比较缓慢, 而且长时值的音符占有较大的比重. 为此, 构建规则模型 R_d , 如式 (4)~(6) 所示.

$$DC(i) = D(i)/D(i-1) \in [2^{-a}, 2^a] \quad (4)$$

$$a = \begin{cases} 2, & \text{若 } D(i) \in S_p, D(i-1) \in S_p \\ 3, & \text{若 } D(i) \in S_p, D(i-1) \notin S_p \end{cases} \quad (5)$$

$$N_l \in \{1, 2\}, \text{ 若 } D(i) \in S_p, D(i-1) \in S_p \quad (6)$$

式 (4) 中, $DC(i)$ 为相邻音符时值的对比度, $D(i)$ 为第 i 个音符的时值; N_l 为知觉统一体内长时值音符的个数. 下文中的时值变异就是在规则模型 R_d 的约束下进行的.

1.3 紧张度

由于“柔和”有“柔软”与“和谐”的意思, 而“和谐”与“协和”意思相近, 因此柔和乐曲的紧张度往往比较低. 依此推理, 需要在算法设计时考虑:

尽量使用协和音程 (即振动频率之比的公倍数较小的音程, 例如纯一、四、五、八度和大小三、六度), 少用或不使用不协和音程 (即振动频率之比的公倍数较大的音程, 例如大小二、七度). 但是过多的使用协和音程, 会在听觉上产生“单一”、“不丰富”的感觉, 这一点无论在和声音程还是旋律音程中都适用, 因为即使声音的物理刺激停止后, 大脑中仍保留着它的听觉印象. 而且这一考虑也与音高律动规则 R_p 产生一定的矛盾. 为了兼顾音乐的美感和表现的对象以及 R_p , 这里设定一个“统一规则” R_u , 如式 (7)~(10) 所示.

$$I_b \in \{0, 2, 3, 4\} \quad (7)$$

$$I_{bb} \in \{0, 2, 3, 4, 5, 7\} \quad (8)$$

$$I_{\rightarrow p} \in \{7, 9, 10, 12\} \quad (9)$$

$$N_{i=1} \in \{1, 2\} \quad (10)$$

其中, I_b , I_{bb} , $I_{\rightarrow p}$ 分别表示小节之内、小节之间、推向高潮对应的音程样本值, $N_{i=1}$ 为一个乐段中小二度音程 (半音数为 1) 的次数, 之所以限定它的使用次数不超过 2, 是因为该音程具有强烈的紧张度.

1.4 音强和起音速度

音强即声音的强弱, 也就是振动体振动幅度的大小. 太强的音响会使人产生强烈的情绪, 而太弱的音响需要较高的注意力, 会使人产生紧张感, 因此柔和乐曲需要中等强弱的音响来表现. 起音速度表达的是一个振动体从开始振动到最大振幅的快慢程度. 由于起音速度过快会使人产生硬、凶的感觉, 因此柔和的乐曲对应的是慢速的起音, 即在演奏时触键的速度要慢. 由于音强和起音速度特征的表现主要靠演奏来完成, 为了给听众或音乐治疗中的患者提供更多二度创作的机会, 以让他们享受到成就感, 提高治疗效果, 这里不再设定具体的规则, 即在谱面上不再呈现描述与音强和起音速度有关的关键词.

2 旋律自动生成算法设计

无论人工作曲还是计算机作曲, 所面对的最基本任务实际上都是如何生成音符序列的问题, 而音符中含有两方面的基本信息: 一是音的高低 (音高信息), 二是音的长短 (时值信息), 因此音符序列可以用并行的音高序列和时值序列来表达, 所以如何生成音符序列的问题也就可以转换为如何生成音高序列和与之并行的时值序列的问题. 对于时值序列的自动生成, 本文采用先初始化后变异的方式, 即先将时值序列的各元素用 1 拍 (本文中时值均以“拍”为单位) 来初始化, 然后通过拆分、合并、转移等运算

对原来的时值序列进行变异,生成新的时值序列,为音高序列搭建起旋律的骨架.对于音高序列的自动生成是在时值序列的基础上,通过操作音高旋律元来完成.下面先来介绍旋律元的定义.

2.1 音高旋律元的定义

以往的作曲算法,其基本操作单位基本都是音符,即“算法直接管理音符”,有点像“司令直接管理士兵”,为了实现“司令管理军官”的目标,这里定义另一种基本操作单位—音高旋律元(Pitch melody unit, PMU).

音高旋律元 PMU 的定义,分为两种情况:一阶 PMU 和二阶 PMU.

将含 1 个音符的 PMU 定义为一阶 PMU (记为 PMU_1).为了兼顾音高的易读性和易处理性,这里说明两种表示方法:一是手写码表示法(记为 P_h);二是 MIDI 码表示法(记为 P_m).

手写码是从人的易读性来考虑的,对应的是以首调唱名法为基础的表示方法,以前面冠以负号表示低音,前面冠以 1 表示高音.考虑到可歌唱性及绝大多数人的音域范围,这里取最低音为 -3 ,最高音为 13 ,用所选取的此范围内的 14 个自然音级个体组成的集合作为手写码的音高论域,即 $P_h = \{-3, -5, -6, -7, 1, 2, 3, 4, 5, 6, 7, 11, 12, 13\}$ (由于 fa 为偏音,mi 与 fa 为小二度音程,会产生较大的紧张度,本着尽量减少紧张度的原则,低音的 fa,即 -4 ,未列入 P_h 中;另外,休止符 0 要单独处理,也不作为该集合的元素).

MIDI 码是从计算机易处理的角度来考虑的,对应的是这 14 个音高个体的 MIDI 音高,可表示为

$$P_m = \{P | P = M + C_k, M \in P_c\} \quad (11)$$

本文在处理 PMU_1 时用 P_m 表示.式(11)中, P 为绝对音高(也用 MIDI 码来表示,本文所有公式中涉及的音高均为 MIDI 码), P_c 为调号为 C 调时 P_h 对应的 MIDI 音高的集合,可表示为 $P_c = \{52, 55, 57, 59, 60, 62, 64, 65, 67, 69, 71, 72, 74, 76\}$, C_k 为 $[-2, 9]$ 中的整数,对应 12 种调号的编码(虽然音乐中有 15 种调号,但有 3 个是等音调).休止符不参与调号的叠加,直接将 P 赋值为 0. PMU_1 一般对应长时值的音符.

将 P_m 中两个元素按照一定音程关系构成的音高序偶定义为二阶 PMU (记为 PMU_2),对应短时值的音符,可由 PMU_1 和 R_u 来完成.考虑到柔和特性,这里设定音程在纯五度之内,如式(12)所示:

$$PMU_2 = \{(P_i, P_j) | \Delta P = |P_i - P_j| \leq 7\} \quad (12)$$

2.2 音高旋律元转换表自动生成的模型构建

生成音高旋律元转换表(Pitch melody unit transition table, PMUTT) 的目的是为旋律计算提供基础数据,它表达的是相邻 PMU 的链接概率.首先根据柔和乐曲的特点,自动生成一阶 PMUTT,然后由一阶 PMUTT 自动生成二阶 PMUTT.

一阶 PMUTT 的自动生成的模型可表达如下:

$$F_1(i, j) = \frac{24}{|PMU_1(i) - PMU_1(j)| + 2} \quad (13)$$

$$P_1(i, j) = \frac{F_1(i, j)}{\sum_{x=0}^{14} F_1(i, x)} \quad (14)$$

式(13)中, $F_1(i, j)$ 为一阶 PMUTT 矩阵中第 i 行、第 j 列的链接频度值, $PMU_1(i)$ 和 $PMU_1(j)$ 分别为第 i 行、第 j 列的 PMU_1 .式(14)中, $P_1(i, j)$ 为链接概率.用式(13)和式(14)来表达一阶 PMUTT 的自动生成模型是基于这样的考虑: $PMU_1(i)$ 和 $PMU_1(j)$ 的音程与链接概率呈相反变化关系,即如果两个 PMU_1 的音程较小,则其链接的概率就较大,它是在规则 R_p 的约束之下,根据柔和乐曲的音高特征,结合十二平均律给出的.

二阶 PMUTT 的自动生成要结合节奏型特征来描述.节奏型可概括抽象为三类,即时值等分型、前长后短型、前短后长型,相应的链接概率计算模型如表 1 所示.

表 1 中, P_{1i} 为第 i 行的 PMU_2 中两个 PMU_1 的链接概率, P_{1ij} 为第 i 行的 PMU 中第 2 个音高和第 j 列的 PMU 中第 1 个音高的链接概率, P_{1j} 为第 j 列的 PMU_2 中两个 PMU_1 的链接概率. x_i 和 y_i ($i = 1, 2, 3$) 分别为行数和列数.通过表 1 可以生成三种类型的 PMUTT 矩阵.

2.3 旋律计算

旋律计算是通过操作 PMU 数据和时值序列生成旋律的行为,这里通过 4 个算子来完成,分别用来生成新 PMU、乐节、乐句、乐曲.生成的旋律的拍号为 4/4(以四分音符为 1 拍,每个小节为 4 拍).各个算子的描述如下.

2.3.1 算子 1 (记为 f_1): 生成 1 个新的 PMU

输入参数为 PMU_i 和 PMUTT (M 行、 N 列);输出参数为 PMU_o ;计算过程如下:

1) 根据式(15)求得链接概率的和矩阵 PMUTTS:

$$S(i, j) = \sum_{x=0}^j P(i, x) \quad (15)$$

表 1 链接概率计算
Table 1 Link probability computing

节奏型	标记符号	PMU 链接方式	链接概率的计算方法
等分	$PMUTT_{1111}$	$PMU_1 + PMU_1$	$P_{11}(i, j) = P_1(i, j)$
等分	$PMUTT_{1111}$	$PMU_2 + PMU_2$	$P_{22}(i, j) = 0.3P_{1i}(x_1, y_1) + 0.4P_{1ij}(x_2, y_2) + 0.3P_{1j}(x_3, y_3)$
前长后短	$PMUTT_{211}$	$PMU_1 + PMU_2$	$P_{12}(i, j) = 0.6P_{1ij}(x_2, y_2) + 0.4P_{1j}(x_3, y_3)$
前短后长	$PMUTT_{112}$	$PMU_2 + PMU_1$	$P_{21}(i, j) = 0.4P_{1i}(x_1, y_1) + 0.6P_{1ij}(x_2, y_2)$

其中, i 和 j 分别为行、列变量, $i \in [0, M - 1]$, $j \in [0, N - 1]$; $S(i, j)$ 为 PMUTTS 中第 i 行第 j 列的概率累计值; $P(i, x)$ 为 PMUTT 中第 i 行第 x 列的概率值.

2) 由式 (16) 和式 (17) 分别获取 PMU_i 在 PMUTTS 中的行数 r 和随机数 R :

$$r = Find(PMUTTS, PMU_i) \quad (16)$$

$$R = Rand[0, S(r, N - 1)] \quad (17)$$

式 (16) 中, $Find$ 为查找函数, 其功能是查找 PMU_i 在 PMUTTS 中的行数; 式 (17) 中, $Rand$ 为随机数生成函数, 该随机数的最小值为 0, 最大值为 $S(r, N - 1)$.

3) 如果 $S(r, x) \geq R$ 且 $\exists j = \min(x)$, 那么 $PMU_o = PMU(j)$, 其中 x 和 j 均为列变量, 即如果 PMUTTS 中第 r 行第 x 列的值大于等于随机数 R , 则取其中列号最小的 PMU 作为输出.

2.3.2 算子 2 (记为 f_2): 生成 1 个乐节的时值序列和音高序列

为了产生柔和的节奏, 这里定义三种时值变异的算子, 分别为拆分、合并和转移. 拆分算子如式 (18)~(21) 所示; 合并算子如式 (22) 所示; 转移算子如式 (23)~(24) 所示. 通过时值变异可将一个肯定、果断的节奏型转化为一个柔和、流畅的节奏型.

$$P_{sp} = \{(N_{sp}, M_{sp}) | N_{sp} \in U_m, M_{sp} \in U_n\} \quad (18)$$

$$U_m = \{1, 2, 3\}, \quad U_n \in \{0, 1\} \quad (19)$$

$$D_{sp}(i) = C(i) \cdot D_{ori}, \quad C_{sp} = \frac{C(i)}{C(i+1)} \quad (20)$$

$$C_{sp} \in \{3, 1\}, \quad i \in \{1, 2\} \quad (21)$$

$$D_{con} = \sum_{i=1}^{N_{con}} D(i), \quad N_{con} \in \{2, 3, 4\} \quad (22)$$

$$\vec{D}_{new} = (1 - 0.25 \cdot n) \vec{D}_{ori}, \quad n \in \{1, 2, 3\} \quad (23)$$

$$\overleftarrow{D}_{new} = \overleftarrow{D}_{ori} + 0.25 \cdot n \cdot \overleftarrow{D}_{ori}, \quad n \in \{1, 2, 3\} \quad (24)$$

式 (18) 中, P_{sp} 为拆分参数, 它是拆分个数 N_{sp} 和拆分方式 M_{sp} 的二元关系集, M_{sp} 表达的是连续拆

分还是相间拆分. 式 (20) 中, $D_{sp}(i)$ 为将 D_{ori} 拆分后的其中一部分, $C(i)$ 为拆分系数, C_{sp} 为拆分比值. 式 (22) 中, D_{con} 为合并后的时值, N_{con} 为合并的个数. 式 (23) 和式 (24) 为转移运算, 其中 \vec{D}_{ori} 、 \vec{D}_{new} 、 \overleftarrow{D}_{ori} 、 \overleftarrow{D}_{new} 分别为时值转出方原始时值、转出方新的时值、转入方原始时值、转入方新的时值.

1) 将该乐节的时值序列集合的任意元素初始化为 $D(i) = 1$, 先进行一次拆分运算, 生成新的时值序列为 D_{sp} , 再根据规则 R_d 确定是进行转移运算还是合并运算, 然后再判断是否满足 R_d , 从而决定下一步的运算方式. 经过几轮操作, 最后生成满足 R_d 的新的时值序列. 由于乐节的结束音和非结束音在特征上存在较大差异, 因此结束音要单独处理. 结束音的特点是稳定、时值较长、以调式主音为常见, 尤其是乐曲最末的音. 考虑到柔和乐曲特征, 本文采用小调式, 即调式主音为 la, 其时值取两个值, 2 拍或 4 拍. 通过这些操作, 便可为音高序列的自动生成搭建了骨架.

2) 这一步的工作是由时值序列和初始的 PMU 自动生成音高序列. 输入参数为 1) 中输出的时值序列 $D(i)$, 起始 PMU_s 和 PMUTT; 输出为该小节的音高序列 $P(i)$; 计算过程如下: 根据 $D(i)$ 中前后相邻时值的比例确定节奏型, 利用表 1 的计算方法和 PMU_s 以及 f_1 生成第 2 个 PMU, 记为 $PMU(2)$, 再根据 $PMU(2)$ 、PMUTT 和 f_1 生成第 3 个 PMU, 依次类推. 在使用 f_1 时, 选用怎样的链接方式取决于当前时值的前后比例. 由于乐节的结尾要体现出句逗的感觉, 这里采用调式主音 la 或属音 mi 来替换算法中得到的音高 (用 la 还是 mi 取决于前一个 PMU 中末尾的音高 P , 哪个与 P 的音程小, 就取哪个). 最后输出由若干个 PMU 构成的音高序列.

2.3.3 算子 3 (记为 f_3): 生成 1 个乐句

乐句一般包括 2 个乐节, 相对于乐节, 乐句的结束音往往时值更长, 因此设计该算子要考虑的问题主要包括: 乐节之间的 PMU 如何连接, 以及乐句结束音的时值和音高处理问题. PMU 如何连接涉及的问题实际上是这两个乐节的关系问题, 这里采用变化重复的旋律发展手法, 即这两个乐节的起始 2 拍

和 PMU 是相同的, 其他序列由算法本身自然发展. 该算子的输入参数为 1 个乐节起始的 PMU_s 以及 PMUTT; 输出为 1 个乐句的 $D(i)$ 和 $P(i)$; 计算过程如下: 通过 PMU_s 、PMUTT 以及 f_2 生成第 1 个乐节, 再用相同的方法生成第 2 个乐节 (这样就保证了这两个乐节的起始 2 拍和 PMU 是相同的, 而后面的序列往往是不同的, 这是由 PMUTT 中数据和随机数与时间有关的双重规则共同决定的). 将第 2 个乐节中的末尾的小节的时值用 4 拍来替换, 音高不变. 输出这两个乐节的时值序列和由若干个 PMU 构成的音高序列.

2.3.4 算子 4 (记为 f_4): 生成 1 首乐曲

乐曲一般包括 1~3 个乐段, 这里取 3 个乐段, 分别为起始、发展、高潮. 该算子的输入参数为起始乐段、发展乐段、高潮乐段对应的起始 2 拍, 以及 PMU, 分别记为 PMU_{1s} , PMU_{2s} , PMU_{3s} (这些参数都是通过对话框人机交互输入的); 输出为整首乐曲的时值序列 $D(i)$ 和音高序列 $P(i)$; 计算过程如下: 通过 PMU_{1s} 、PMUTT 以及 f_3 生成起始乐段的 2~3 个乐句. 通过 PMU_{2s} 、PMUTT 以及 f_3 生成发展乐段的 2~3 个乐句. 通过 PMU_{3s} 、PMUTT 以及 f_3 生成高潮乐段的 2~3 个乐句. 最后输出由这 3 个乐段的时值序列和音高序列.

容易看出, 乐曲的生成是这 4 个算子对旋律元进行操作和对时值进行变异综合作用的结果, 下面通过实验来验证该算法.

3 实验结果及对比分析

为了描述的方便, 将本文的算法记为 “Model-PMU-DurationVariation” (即 “模型-音高旋律元-时值变异” 算法, 简称为 “M-PMU-DV” 算法), 将要对比的通过分析已有乐曲的以音符为基本操作单位的马尔可夫模型方法记为 “Imitation-Note-Markov” 算法 (即 “模仿-音符-马尔可夫” 算法, 简称为 “I-N-M” 算法).

考虑到柔和乐曲的特点, 在两种算法中设定的乐曲参数值均为: 拍号: 4/4; 调式: 小调; 调号: 1 = F; 速度: 75 拍/分钟; 小节数: $N = 24$. 实验平台为 VC++.

3.1 M-PMU-DV 算法的作曲实验

为了设定乐曲的结构参数, 这里设计了一个交互界面, 如图 1 所示. 输入的各乐段的起始 2 拍和 PMU (起始音高元) 依次为 “1.5, 0.5”, “-6, 1”, “1, 1”, “1, 1”, “1, 0.5, 0.5”, “6”. 创作 1000 首乐曲所需的时间为 5.3 秒, 每首曲子的文件名格式为 “第 * 首 (旋律元为单位).mid”, 随机选择了其中的 3 首, 其文件名分别为 “第 42 首 (旋律元为单位)”、“第 335 首 (旋律元为单位)” 和 “第 898 首 (旋律元为单位)”, 其乐谱分别如图 2(a)、图 3(a) 和图 4(a) 所示 (可登录 <http://19189377.5sing.com/> 或 <http://caoxizhengzuoqu.space.a8.com/>, 试听它们对应的部分 mp3 乐曲).

图 1 乐曲结构设置界面

Fig. 1 Interface of music structure setting

(a) M-PMU-DV 算法
(a) M-PMU-DV algorithm

(b) I-N-M 算法
(b) I-N-M algorithm

图 2 第 42 首乐曲的乐谱
Fig. 2 Scores of the 42nd music

(a) M-PMU-DV 算法
(a) M-PMU-DV algorithm

(b) I-N-M 算法
(b) I-N-M algorithm

图 3 第 335 首乐曲的乐谱
Fig. 3 Scores of the 335th music

(a) M-PMU-DV 算法
(a) M-PMU-DV algorithm

(b) I-N-M 算法
(b) I-N-M algorithm

图 4 第 898 首乐曲的乐谱
Fig. 4 Scores of the 898th music

3.2 I-N-M 算法的作曲实验

为了与 M-PMU-DV 算法进行对比分析, 下面进行 I-N-M 算法的作曲实验, 其实现过程如下:

1) 选取要模仿的乐曲: 考虑到柔和, 选取《故乡的原风景》, 其乐谱如图 5 所示.



图 5 《故乡的原风景》的乐谱

Fig. 5 Scores of "Hometown of the scenery"

2) 分析原作品: 以音符为单位统计图 5 中音高的转移频度, 并进行横向累计, 如表 2 所示

表 2 音高转移频度的横向累计

Table 2 Row sums of transition frequency of pitches

图 5 中的音高论域	0	-5	-6	1	2	3	5	6
0	2	3	3	3	3	3	3	3
-5	0	0	4	4	4	4	4	4
-6	0	0	0	4	4	7	8	8
1	0	0	4	4	10	12	13	13
2	0	0	0	7	7	12	12	12
3	0	3	3	3	11	13	14	14
5	0	0	0	0	0	3	5	10
6	1	1	1	1	1	1	6	6

(音高用手写码表示); 用同样的方法以小节为单位进行时值序列转移频度的横向累计, 如表 3(时值用拍数表示, 该表为时值序列的编码表) 和表 4 所示.

3) 创作乐曲: 将第 1 个音符的音高初始化为表 2 中某一行的音高值, 在 $1 \sim C_8$ 之间产生一个随机数 (C_8 为该行最后一列的数值), 该行第 $1 \sim 8$ 列中第 1 个大于或等于该随机数的列所对应的音高值即为下一个音高值, 依次循环即可生成音高序列. 用同样的方法即可生成时值序列. 根据设定的小节数 24 即可完成一首乐曲的创作. 为了与 M-PMU-DV 算法比较, 这里也创作 1000 首, 每首曲子的文件名格式为“第 * 首 (音符为单位).mid”, 这里也取 3 首, 其文件名分别为“第 42 首 (音符为单位)”、“第 335 首 (音符为单位)”和“第 898 首 (音符为单位)”, 其乐谱分别如图 2(b)、图 3(b) 和图 4(b) 所示.

下面对算法的正确性和复杂度进行对比分析.

表 3 小节时值序列的编码

Table 3 Coding of duration array of every bar

图 5 中小节的时值序列论域	编码	图 5 中小节的时值序列论域	编码
1, 1, 0.5, 0.5, 0.5, 0.5	1	1.5, 0.5, 1, 1	8
2.5, 0.25, 0.25, 0.75, 0.25	2	0.25, 0.25, 2.5, 0.5, 0.25, 0.25	9
2.5, 0.5, 0.5, 0.5	3	1.5, 0.5, 1, 0.5, 0.5	10
1.5, 0.5, 1, 0.5, 0.25, 0.25	4	0.25, 0.25, 2.5, 0.75, 0.25	11
2.5, 0.25, 0.25, 1	5	3, 0.5, 0.5	12
3, 1	6	0.25, 0.25, 2, 0.25, 0.25, 1	13
3, 0.5, 0.25, 0.25	7	4	14

表 4 时值序列转移频度的横向累计
Table 4 Row sums of transition frequency of duration arrays

时值序列编码	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	2	3	3	3	3	3	3	3	3	3	3	3
3	0	1	1	1	2	2	2	2	2	2	2	2	2	2
4	0	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	0	1	1	1	1	1	1	1	1	1
6	0	0	0	0	0	0	1	1	1	1	1	1	1	1
7	0	0	0	0	0	0	0	1	1	1	1	1	1	1
8	0	0	0	0	0	0	0	0	1	1	1	1	1	1
9	0	0	0	0	0	0	0	0	0	1	1	1	1	1
10	0	0	0	0	0	0	0	0	0	0	1	1	1	1
11	0	0	0	0	0	0	0	0	0	0	0	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	1	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1

3.3 算法的正确性对比

衡量作曲算法的正确性应该从其生成的乐曲的质量水平来考虑. 考虑到音乐是一个感性和理性的统一体, 这里采用客观评价结合主观评价的方法, 表 5~表 7 分别列出了客观评价结果和主观评价结果, 其中表 7 中的评价值是由 30 人对每首乐曲综合评价分数的平均值 (评价前, 先将图中的乐谱转换为 MIDI 文件, 然后转换为 mp3), 参评人员包括作曲专业人士 10 人、非专业人士 20 人, 综合评价 = 专业人士评价的平均值 * 0.6 + 非专业人士评价的平均值 * 0.4. 在表 5 中美感度与柔和的明确度均量化为 100 份.

从图 2(a)、图 3(a) 和图 4(a) 可以看出, 乐节之间为变化重复关系, 乐句之间为贯穿关系, 乐段之间为自由引申关系, 从第 17 小节开始为高潮. 从图 2(b)、图 3(b) 和图 4(b) 可以看出, 乐句不明显, 关系松散, 长时值的音处可近似看成乐句的结束, 少量近似乐句间存在模糊的贯穿关系; 高潮不明显, 在计算高潮的体现度上近似从第 17 小节开始.

从表 6 的指标可以得出, M-PMU-DV 算法与 I-N-M 算法相比, 在相邻乐节或乐句的关系相关度上提高了 $(6 - 2)/2 = 2$ 倍, 使得乐曲逻辑严密程度得到了大幅提升; 在乐句结束音位置的稳定度上提高了 $(1 - 0.29)/0.29 = 2.45$ 倍, 使得乐曲结构更严谨; 在高潮的体现度上, M-PMU-DV 算法对推向高潮、情感的上升具有良好的效果, 而 I-N-M 算法会让听众感受不到乐曲高潮的存在; 在音程符合度上, 提高了 $(98.4\% - 87.4\%)/87.4\% = 0.13$ 倍, 这

个数据基本不能说明什么问题, 因为 I-N-M 算法是建立在原有乐曲上, 音程大小不会出现过多的突变; 在“坏点”音符的百分比上, M-PMU-DV 算法由于自身的控制严格, 因此没有坏点, 而 I-N-M 算法由于结构松散, 存在一定的坏点. 从表 5 的指标可以得出, M-PMU-DV 算法创作的乐音与 I-N-M 算法创作的乐音相比, 在美感度上提高了 $(82.2 - 40.6)/40.6 = 1.02$ 倍, 在柔和的明确度上, 提高了 $(88.6 - 53.7)/53.7 = 0.65$ 倍.

3.4 算法的复杂度对比

表 8 列出了时间复杂度和空间复杂度的对比, 其中耗时数据是在同一台机器上运行时得到的, 配置为 Windows XP, CPU i3-2100, 内存 4G.

从表 8 可以看出, M-PMU-DV 算法与 I-N-M 算法相比, 在空间占有方面, 两者基本持平, 但耗时较小.

4 总结和展望

本文从目前自动作曲的研究现状入手, 结合马尔可夫模型在作曲中存在的问题, 面向柔和乐曲, 提出了一种基于音高旋律元的作曲算法. 首先构建了算法的设计规则模型, 然后综合考虑了音高的逻辑性和灵活性的特点, 定义了一阶和二阶 PMU, 并根据节奏型特点, 抽象并构建了三类 PMUTT: $PMUTT_{1111}$, $PMUTT_{211}$ 和 $PMUTT_{112}$. 在构建旋律计算的算法时, 设计了 4 个算子, 分别用来生成新的 PMU、乐节、乐句和乐曲. 在对时值进行变异时, 提出了拆分、合并和转移运算的模型.

表 5 客观评价指标
Table 5 Objective evaluation indexes

指标	指标的求解策略
相邻乐节或乐句的关系相关度 R	如果相邻的乐节或乐句在关系上存在着重复 (完全重复或变化重复)、模进 (严格模进或自由模进)、对比、自由引申、承递、贯穿等其中的一种, 则称乐节或乐句相关. R 的数值用具有相关性的乐句的个数来表示
乐句结束音位置的稳定度 S	乐句中包含的最少的小节数和最多的小节数的比值
高潮的体现度 P	乐曲黄金分割点后的平均音高减去黄金分割点前的平均音高 (计算时采用 MIDI 码)
音程符合度 I	考虑到柔和乐曲的音程特征, 如果乐句内相邻两个音符的音程在 4 度以内, 则该音程可判定为符合音程 I 可用满足符合音程的音符百分比来表示
“坏点” 音符的百分比 B	如果乐句内相邻两个音符的音程在 8 度以上, 则后面的音符判定为“坏点” 音符; 同时, 如果某休止符的出现没有对旋律的发展起到推动作用, 也没有起到句逗的作用, 则该休止符也判定为“坏点” 音符

表 6 客观评价结果对比
Table 6 Comparison of the objective evaluation results

指标	M-PMU-DV 算法				I-N-M 算法			
	第 42 首 (图 2 (a))	第 335 首 (图 3 (a))	第 898 首 (图 4 (a))	综合评价	第 42 首 (图 2 (b))	第 335 首 (图 3 (b))	第 898 首 (图 4 (b))	综合评价
R	6	6	6	6	2	2	2	2
S	$4/4 = 1$	$4/4 = 1$	$4/4 = 1$	1	$3/9 = 0.33$	$2/9 = 0.22$	$2/6 = 0.33$	0.29
P	$73.1-64.4 = 8.7$	$72.8-63.5 = 9.3$	$75.2-62.8 = 12.4$	10.13	$62.1-61.6 = 0.5$	$67.4-66.9 = 0.5$	$61.7-62.6 = -0.9$	0.03
I	$99/102 = 97.1\%$	$101/102 = 99.0\%$	$98/99 = 99.0\%$	98.4%	$88/104 = 84.6\%$	$100/112 = 89.3\%$	$91/103 = 88.3\%$	87.4%
B	0	0	0	0	$5/104 = 4.8\%$	$1/112 = 0.9\%$	$6/103 = 5.8\%$	3.8%

表 7 主观评价结果对比
Table 7 Comparison of the subjective evaluation results

指标	M-PMU-DV 算法				I-N-M 算法			
	第 42 首 (图 2 (a))	第 335 首 (图 3 (a))	第 898 首 (图 4 (a))	综合评价	第 42 首 (图 2 (b))	第 335 首 (图 3 (b))	第 898 首 (图 4 (b))	综合评价
美感度	85.6	79.2	81.8	82.2	38.9	42.7	40.3	40.6
柔和的明确度	88.9	86.7	90.3	88.6	51.7	60.4	49.1	53.7

表 8 算法复杂度对比
Table 8 Comparison of algorithmic complexity

算法	时间复杂度及耗时	空间复杂度
M-PMU-DV 算法	计算量为 $C_{PMU} = C_P + C_D$, 其中 C_P 、 C_D 分别为操作 PMU 和时值变异的计算量, 生成 1 首乐曲平均耗时 5.3 毫秒	$S_{PMU} = S_{PMUd} + S_{PMUTT} + S_{PMUo}$, 其中 S_{PMUd} 为 PMU 占的空间, S_{PMUTT} 为 PMUTT 占的空间, S_{PMUo} 为操作函数占的空间
I-N-M 算法	计算量为 $C_M = C_1 + C_2$, 其中 C_1 为统计已有乐曲音符概率的计算量, C_2 为操作音符的计算量, C_2 接近于 $2(C_P + C_D)$, 生成 1 首乐曲平均耗时 11.8 毫秒	$S_N = S_{Nd} + S_{No}$, 其中 S_{Nd} 为音符数据的占的空间, S_{No} 为操作函数占的空间 (S_{Nd} 和 S_{PMUd} 近似相等, S_{No} 一般大于等于 $2S_{PMUo}$, 但由于 S_{PMUTT} 的存在, 使得 S_{PMU} 和 S_N 近似相等)
对比结果	节省了大约一半的时间	基本持平

多次实验分析发现, M-PMU-DV 算法与以音符为操作单位以及输入为已有作品的算法相比, 其优点主要有:

1) 由于音高旋律元是一个“预制的构件”, 因此操作相对简单, 而且它本身已经有一定的含义, 所以由其生成的旋律的逻辑会更严密, 对乐思的表达会更深刻, 即用一个比较简单的算法生成具有一定含义的乐曲; 而以音符为基本操作单位的算法中, 由于音符个体不具有任何含义, 因此由其生成的旋律往往会缺乏逻辑性和章法. 与马尔可夫模型相比, M-PMU-DV 算法解决了 2 维马尔可夫模型的随意性问题, 同时也克服了 n 维马尔可夫模型的难以管理问题.

2) 该算法构建了自动生成音高旋律元转换表的模型, 实现了基础数据获取的自动化. 该方法与传统的对大量作品分析从而获取转换表数据的方法相比, 省略了由分析大量作品而产生的庞大工作量. 与“输入已有作品, 输出仿真的作品的创作模式”相比, 克服了以偏概全问题. 同时由于乐曲的起始旋律元是人工设置的, 这样既实现了自动化, 又保证了一定程度的灵活性.

3) 由该算法生成的是乐曲, 充分考虑了乐曲的结构和乐曲单元的组织的具体问题, 在组织过程中始终以柔和这种确定的风格为驱动, 即面向的对象非常明确; 而现存的其他算法生成的绝大多数是短小的旋律片段, 或是简单的几个乐句等, 而且面向的对象往往不够明确.

本文的算法还有不足之处需要进一步研究: 该算法在生成音符密集的旋律时会表现出局限性, 因为音符密集意味着一个小节的音符个数较多, 导致音高旋律元转换的次数较多, 这样就使得一个小节的音高线的逻辑性向着松散的方向发展. 要解决这个问题, 可考虑将旋律元在音符数目的规模上扩大, 这样就使得它的个数有所减少, 但过大的旋律元会影响创作的灵活性, 因此要兼顾这两个问题, 设计规模适宜的音高旋律元.

References

- Shin H S, Kim J H. Music therapy on anxiety, stress and maternal-fetal attachment in pregnant women during transvaginal ultrasound. *Asian Nursing Research*, 2011, **5**(1): 19–27
- Korhan E A, Khorshid L, Uyar M. The effect of music therapy on physiological signs of anxiety in patients receiving mechanical ventilatory support. *Journal of Clinical Nursing*, 2011, **20**(7–8): 1026–1034
- Streeter E, Davies M E P, Reiss J D, Hunt A, Caley R, Physicist C, Roberts C, Therapy D M. Computer aided music therapy evaluation: testing the music therapy log-book prototype 1 system. *The Arts in Psychotherapy*, 2012, **39**(1): 1–10
- Feng Yin, Zhou Chang-Le. Advances in algorithmic composition. *Journal of Software*, 2006, **17**(2): 209–215 (冯寅, 周昌乐. 算法作曲的研究进展. *软件学报*, 2006, **17**(2): 209–215)
- Alpern A. Techniques for Algorithmic Composition of Music [Online], available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.23.9364&rep=rep1&type=pdf>, November 3, 2011
- Basharin G P, Langville A N, Naumov V A. The life and work of A. A. Markov. *Linear Algebra and Its Applications*, 2004, **386**(7): 3–26
- Verbeugt K, Dinolfo M, Fayer M. Extracting patterns in music for composition via Markov chains. In: *Proceedings of the 17th International Conference on Innovations in Applied Artificial Intelligence*. Berlin: Springer-Verlag, 2004. 1123–1132
- Deng Xin-Yang, Deng Yong, Zhang Ya-Juan, Liu Qi. A belief Markov model and its application. *Acta Automatica Sinica*, 2012, **38**(4): 666–672 (邓鑫洋, 邓勇, 章雅娟, 刘琪. 一种信度马尔科夫模型及应用. *自动化学报*, 2012, **38**(4): 666–672)
- Niu Jian-Jun, Deng Zhi-Dong. Markov chain-based distributed scheduling approach for wireless sensor network. *Acta Automatica Sinica*, 2010, **36**(5): 685–695 (牛建军, 邓志东. 基于马尔可夫链的无线传感器网络分布式调度方法. *自动化学报*, 2010, **36**(5): 685–695)
- Oliwa T, Wagner M. Composing music with neural networks and probabilistic finite-state machines. In: *Proceedings of the 2008 Conference on Applications of Evolutionary Computing*. Berlin: Springer-Verlag, 2008. 503–508
- Liu X F, Tse C K, Small M. Complex network structure of musical compositions: algorithmic generation of appealing music. *Physica A: Statistical Mechanics and its Applications*, 2010, **389**(1): 126–132
- Miranda E R, Biles J A. *Evolutionary Computer Music*. Berlin: Springer-Verlag, 2007. 1–26
- McDermott J, O'Reilly U M. An executable graph representation for evolutionary generative music. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2011. 403–410
- Bell C. Algorithmic music composition using dynamic Markov chains and genetic algorithms. *Journal of Computing Sciences in Colleges*, 2011, **27**(2): 99–107
- Tzimeas D, Mangina E. Dynamic techniques for genetic algorithm-based music systems. *Computer Music Journal*, 2009, **33**(3): 45–60
- Jensen J H, Hadow P C. Evolutionary music composition based on Zipf's law. In: *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*. New York, USA: ACM, 2011. 41–42

- 17 Jin Di, Liu Jie, Yang Bo, He Dong-Xiao, Liu Da-You. Genetic algorithm with local search for community detection in large-scale complex networks. *Acta Automatica Sinica*, 2011, **37**(7): 873–882

(金弟, 刘杰, 杨博, 何东晓, 刘大有. 局部搜索与遗传算法结合的大规模复杂网络社区探测. *自动化学报*, 2011, **37**(7): 873–882)

- 18 Huang Xin, Dong Jun. Research and implementation of experience-based computer-aided algorithmic composing. *Mind and Computation*, 2007, **1**(2): 269–285

(黄欣, 董军. 基于经验的计算机作曲方法的研究与实现. *心智与计算*, 2007, **1**(2): 269–285)

- 19 Ariza C. Automata bending: applications of dynamic mutation and dynamic rules in modular one-dimensional cellular automata. *Computer Music Journal*, 2007, **31**(1): 29–49

- 20 Burraston D, Edmonds E. Cellular automata in generative electronic music and sonic art: a historical and technical review. *Digital Creativity*, 2005, **16**(3): 165–185

- 21 Boenn G, Brain M, de Vos M, Ffitch J. Automatic music composition using answer set programming. *Theory and Practice of Logic Programming*, 2011, **11**(2–3): 397–427

- 22 Han Yan-Lin. Research on Possibility Construction Space Theory and Computer Composition Thinking Model [Ph. D. dissertation], Huazhong University of Science & Technology, China, 2010

(韩艳玲. 可能性构造空间理论与计算机作曲思维模型研究 [博士学位论文], 华中科技大学, 中国, 2010)

- 23 Aguilera G, Galán J L, Madrid R, Martínez A M, Padilla Y, Rodríguez P. Automated generation of contrapuntal musical compositions using probabilistic logic in derive. *Mathematics and Computers in Simulation*, 2010, **80**(2): 1200–1211

- 24 Zhou Hai-Hong. *Music and the World It Expresses—an Aesthetic and Psychological Study on Relations between Music Sound and the Objects It Expressed*. Beijing: Central Conservatory of Music Press, 2004. 50–122

(周海宏. 音乐与其表现的世界—对音乐音响与其表现对象之间的关系的心理学与美学研究. 北京: 中央音乐学院出版社, 2004. 50–122)



曹西征 河南师范大学副教授, 博士. 主要研究方向为自动作曲, 计算机音乐技术. 本文通信作者.

E-mail: caoxizheng@126.com

(CAO Xi-Zheng Associate professor at Henan Normal University, Ph. D.. His research interest covers automated

composition and computer music technology. Corresponding author of this paper.)



毛文涛 河南师范大学讲师, 博士. 主要研究方向为人工智能应用, 机器学习.

E-mail: maowt.mail@gmail.com

(MAO Wen-Tao Lecturer at Henan Normal University, Ph. D.. His research interest covers artificial intelligence application and machine learning.)



乔 锟 河南师范大学硕士研究生. 主要研究方向为自动作曲, 计算机音乐技术.

E-mail: qkbeyond@163.com

(QIAO Kun Master student at Henan Normal University. His research interest covers automated composition and computer music technology.)



程小丽 河南师范大学硕士研究生. 主要研究方向为自动作曲, 计算机音乐技术. E-mail: belinlily@126.com

(CHENG Xiao-Li Master student at Henan Normal University. Her research interest covers automated composition and computer music technology.)



蔡会娟 河南师范大学硕士研究生. 主要研究方向为自动作曲, 计算机音乐技术. E-mail: 455843744@qq.com

(CAI Hui-Juan Master student at Henan Normal University. Her research interest covers automated composition and computer music technology.)



温辉明 河南师范大学副教授. 主要研究方向为理论作曲.

E-mail: whm_168_good@163.com

(WEN Hui-Ming Associate professor at Henan Normal University. His main research interest is theory-composition.)